

AD-A189 843

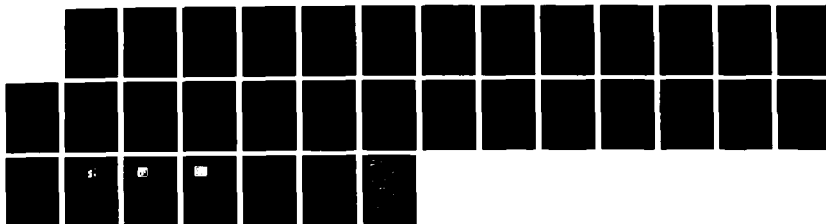
ON RELAXATION ALGORITHMS BASED ON MARKOV RANDOM FIELDS  
(U) ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE  
P B CHOU ET AL. 10 JUL 87 TR-212 N00014-82-K-0193

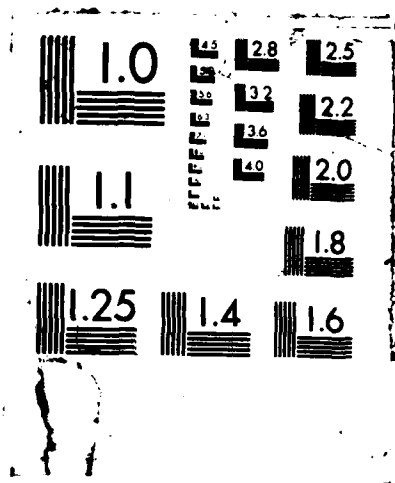
1/1

UNCLASSIFIED

F/G 23/3

ML





AD-A189 043

DTIC FILE COPY

4

On Relaxation Algorithms  
Based on Markov Random Fields\*

Paul B. Chou  
Rajeev Raman  
Computer Science Department  
The University of Rochester  
Rochester, New York 14627

TR 212  
July 10, 1987

SECRET

DTIC  
ELECTE  
JAN 21 1988  
S D

Rochester

Department of Computer Science  
University of Rochester  
Rochester, New York 14627

DISTRIBUTION STATEMENT A  
Approved for public release;  
Distribution Unlimited

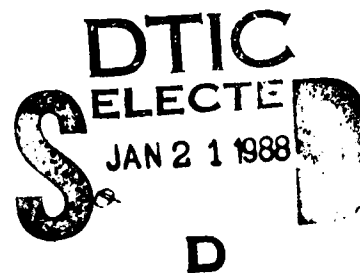
38 1 14 084

4

## On Relaxation Algorithms Based on Markov Random Fields\*

Paul B. Chou  
Rajeev Raman  
Computer Science Department  
The University of Rochester  
Rochester, New York 14627

TR 212  
July 10, 1987

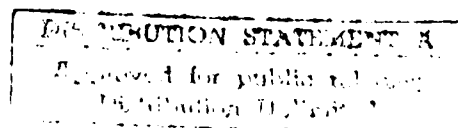


### ABSTRACT

Many computer vision problems can be formulated as computing the minimum energy states of thermal dynamic systems. However, due to the complexity of the energy functions, the solutions to the minimization problem are very difficult to acquire in practice. Stochastic and deterministic methods exist to approximate the solutions, but they fail to be both efficient and robust. In this paper, we describe a new deterministic method - the Highest Confidence First\*\* algorithm - to approximate the minimum energy solution to the image labeling problem under the Maximum A Posteriori (MAP) criterion. This method uses Markov Random Fields to model spatial prior knowledge of images and likelihood probabilities to represent external observations regarding hypotheses of image entities. Following an order decided by a dynamic stability measure, the image entities make local estimates based on the combined knowledge of priors and observations. We show that, in practice, the solutions so constructed compare favorably to the ones produced by existing methods and that the computation is more predictable and less expensive.

\*This work was supported in part by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 under Contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC). This work was also supported in part by U.S. Army Engineering Topographic Laboratories research contract no. DACA76-85-C-0001, in part by NSF Coordinated Experimental Research grant no. DCR-8320136, in part by ONR/DARPA research contract no. N00014-82-K-0193, and in part by a grant from the Eastman Kodak Company. We thank the Xerox Corporation University Grants Program for providing equipment used in the preparation of this paper.

\*\*Also known as "He who hesitates is last".



REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR 212	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  On Relaxation Algorithms Based on Markov Random Fields		5. TYPE OF REPORT & PERIOD COVERED  technical report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)  Paul B. Chou and Rajeev Raman		8. CONTRACT OR GRANT NUMBER(s)  DACA76-85-C-0001 N00014-82-K-0193
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dept. of Computer Science 535 Computer Studies Bldg. Univ. of Rochester, Rochester, NY 14627		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DARPA / 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE July 1987
		13. NUMBER OF PAGES 29
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report)  unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Distribution of this document is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  image segmentation Bayesian-probabilistic approach Markov Random Fields energy minimization		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  Many computer vision problems can be formulated as computing the minimum energy states of thermal dynamic systems. However, due to the complexity of the energy functions, the solutions to the minimization problem are very difficult to acquire in practice. Stochastic and deterministic methods exist to approximate the solutions, but they fail to be both efficient and robust. In this paper, we describe a new deterministic method--the Highest Confidence First algorithm--to approximate the minimum energy solution to the image		

## 20. ABSTRACT (Continued)

labeling problem under the Maximum A Posteriori (MAP) criterion. This method uses Markov Random Fields to model spatial prior knowledge of images and likelihood probabilities to represent external observations regarding hypotheses of image entities. Following an order decided by a dynamic stability measure, the image entities make local estimates based on the combined knowledge of priors and observations. ~~We show that,~~ in practice, the solutions so constructed compare favorably to the ones produced by existing methods and that the computation is more predictable and less expensive. *Keywords:*

*Image Segmentation, Image Annotation*



Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

## 1. Introduction

Probability theory has found many applications in representing the uncertainty of various kinds of knowledge and in reasoning about the world [Feldman and Yakimovsky 1974] [Duda, Hart, and Nilsson 1976] [Peleg 1980] [Pearl 1985]. It appeals to the AI community for many reasons, among which are that it provides a well-developed mathematical theory for using uncertainty measures in making decisions, and that it provides well-known ways of incorporating empirical data. However, there have been few successful attempts at utilizing this tool in practical machine vision systems. It is apparently not because this domain is any the less "uncertain" but because the complexity of the information in this domain hinders the advancement of probabilistic approaches.

It is our goal to demonstrate the practicability of applying the Bayesian-probability formalism to complex domains, such as the image labeling problem discussed in this paper. The *image labeling problem* is to assign labels to image entities such as regions, line segments, and pixels. The set of labels, usually reflecting the photometric and geometrical phenomena of the scene, is mutually exclusive and exhaustive at a particular level of abstraction. Denote the set of labels  $\{l_1, l_2, \dots, l_Q\}$  as  $L$ , and the set of the image entities  $\{s_1, s_2, \dots, s_N\}$  as  $S$ . Any mapping from  $S$  to  $L$  is a *feasible solution* to the labeling problem. To choose an "optimal" solution from the set of feasible solutions -  $\Omega$ , image observations as well as prior knowledge about spatial relations between labels are used to evaluate the goodness of each solution. This work follows the probabilistic model for visual computation proposed in [Chou and Brown 1987b]. Spatial prior knowledge and local visual observations are separately represented in terms of probabilities. This decoupling provides a clean and uniform way of modeling information at different levels of abstraction, and therefore to modularize the design and implementation of probabilistic systems. Bayes' rule is used to combine priors and observations to form the *a posteriori* probabilities representing the updated knowledge. The labeling problem is then formulated as a minimization problem based on the Bayesian decision rationale. We shall show that by using a new algorithm proposed here to estimate the optimal solution to the minimization problem so formulated, it is possible to achieve excellent results with relatively little computation, given a set of reasonable assumptions.

The organization of this paper is as follows. We discuss how to encode the *a priori* knowledge of image events with the Markov random field (MRF) models in Section 2. The Bayesian decision rationale is discussed in Section 3. In Section 4, we review several stochastic relaxation methods that, in principle, could find the optimal solutions given enough computational resources. We describe a new deterministic estimation algorithm in Section 5 that our experiments indicate to be superior to existing methods. Experiments on edge detection - an instance of the labeling problem - with both synthetic and natural images are conducted with an MRF simulation/estimation package implemented by the authors. Results from various estimation schemes are compared in Section 6.

## 2. Markov Random Fields and Gibbs Distributions

Markov Random Fields have been used for image modeling in many applications for the past few years [Hassner and Slansky 1980] [Cross and Jain 1983] [Marroquin, Mitter, and Poggio 1985] [Geman and Geman 1984] [Derin and Cole 1986] [Chou and Brown 1987a]. In this section, we review the properties of MRF's and discuss how to

encode prior knowledge in this formalism. We refer the reader to [Kindermann and Snell 1980] for an extensive treatment of MRF's.

### 2.1. Noncausal Markovian Dependency

Let  $X = \{X_s, s \in S\}$  denote a set of random variables indexed by  $S$ . Without loss of generality, assume all variables in  $X$  have a common state space  $L$ , so that  $X_s \in L$ . Let  $\{X = \omega\}$  be the event  $\{X_{s_1} = \omega_{s_1}, \dots, X_{s_N} = \omega_{s_N}\}$ , where  $\omega = (\omega_{s_1}, \omega_{s_2}, \dots, \omega_{s_N})$ ,  $\omega_s \in L$ , is a configuration of  $X$ . Since a configuration of  $X$  is also a feasible solution to the labeling problem,  $\Omega$  also denotes the set of all possible configurations.

Let  $E$  be a set of unordered pairs  $(s_i, s_j)$ 's representing the "connections" between the elements in  $S$ . The semantics of the connections will become clear shortly.  $E$  defines a neighborhood system  $\Gamma = \{N_s | s \in S\}$ , where  $N_s$  is the neighborhood of  $s$  in the sense that

- (1)  $s \notin N_s$ , and
- (2)  $r \in N_s$  if and only if  $(s, r) \in E$ .

$X$  is a *Markov Random Field* with respect to  $\Gamma$  and  $P$ , where  $P$  is a probability function, if and only if

$$(\text{positivity}) \quad P(X = \omega) > 0 \text{ for all } \omega \in \Omega \quad (2.1)$$

$$(\text{Markovianity}) \quad P(X_s = \omega_s | X_r = \omega_r, r \in S, r \neq s) = P(X_s = \omega_s | X_r = \omega_r, r \in N_s) \quad (2.2)$$

The set of conditional probabilities on the left-hand side of (2.2) is called the *local characteristics* that characterizes the random field. It can be shown that the joint probability distribution  $P(X = \omega)$  of any random field satisfying (2.1) is uniquely determined by these conditional probabilities [Besag 1974]. An intuitive interpretation of (2.2) is that the contextual information provided by  $S - s$  to  $s$  is the same as the information provided by the neighbors of  $s$ . Thus the effects of members of the field upon each other is limited to local interaction as defined by the neighborhood. Notice that any random field satisfying (2.1) is an MRF if the neighborhoods are large enough to encompass all the dependencies.

### 2.2. Encoding Prior Knowledge and Gibbs Distributions

The utility of the MRF concept for image labeling problems is that the prior knowledge about spatial dependencies among the image entities can be adequately modeled with neighborhoods that are small enough for practical purposes. Very often, the image entities are regularly structured and prior distributions on the image are homogeneous and isotropic. In such cases, the number of parameters needed to specify the priors is just a fraction of  $Q^M$ , where  $M$  is the size of the neighborhoods. This is a significant saving over  $Q^N$  - the number of possible configurations, especially when  $M$  is small.

There are difficulties, as stated in [Geman and Geman 1984], associated with using the MRF formulation by itself:

- (1) The joint distribution of the  $X_s$  is not apparent;
- (2) It is extremely difficult to spot local characteristics, i.e., to determine when a given set of functions are conditional probabilities for some distribution on  $\Omega$ .

(1) is not a serious problem for some special classes of MRF models such as *Markov Mesh* (MM) processes [Kanal 1980], since their joint distributions can be represented in a recursive formulation due to the casual dependency assumed. For (2), parametric probability distributions such as Gaussian and binomial, have been used in the literature [Cross and Jain 1983] [Cohen and Cooper 1987]. Using such distributions further simplifies the encoding of the local characteristics and has shown some impressive results on modeling and generating texture patterns. However, whether these kinds of simplifications preserve the power of MRF's for modeling spatial knowledge remains questionable.

Fortunately, these difficulties vanished when the following property of MRF's was realized.

**Hammersley-Clifford Theorem:** A random field  $X$  is an MRF with respect to a neighborhood system  $\Gamma$  if and only if there exists a function  $V$  such that

$$P(\omega) = \frac{e^{-\frac{1}{T}U(\omega)}}{Z} \quad \text{for all } \omega \in \Omega \quad (2.3)$$

where  $T$  and  $Z$  are constants and

$$U(\omega) = \sum_{c \in C} V_c(\omega). \quad (2.4)$$

$C$  denotes the set of totally connected subgraphs (cliques) with respect to  $\Gamma$ .  $Z$  is a normalizing constant and is called the *partition function*.

The probability distribution defined by (2.3) and (2.4) is called a *Gibbs distribution* with respect to  $\Gamma$ . The class of Gibbs distributions has been extensively applied to model physical systems, such as ferromagnets, ideal gases, and binary alloys. When such systems are in a state of *thermal equilibrium*, the fluctuations of their configurations follow a Gibbs distribution. In statistical mechanics terminology,  $U$  is the *energy* function of a system. The  $V_c$  functions represent the *potentials* contributed to the total energy from the local interactions of the elements of clique  $c$ .  $T$ , the *temperature* of the system, controls the "flatness" of the distribution of the configurations.

Gibbs distributions, and therefore MRF's, possess a property that appears to be desirable for modeling - when constrained by a fixed expected value of some sufficient statistic of the random field, the *maximum entropy* distribution among the class of distributions compatible with the constraint is a Gibbs distribution.

The MRF-Gibbs equivalence not only relates the local conditional probabilities to the global joint probabilities, but also provides us a conceptually simpler way of specifying MRF's - specifying potentials. The importance of the joint probabilities will become evident in the next section. The local characteristics can be computed from the potential function through the following relation:

$$P(X_s = \omega_s | X_r = \omega_r, r \neq s) = \frac{e^{-\frac{1}{T} \sum_{c \in C_s} V_c(\omega)}}{\sum_{\omega'} e^{-\frac{1}{T} \sum_{c \in C_s} V_c(\omega')}} \quad (2.5)$$

where  $C_s$  is the set of cliques that contain  $s$ , and  $\omega'$  is any configuration of the field that agrees with  $\omega$  everywhere except possibly  $s$ .

There has been little work that applies statistical estimation methods to estimate parameters used for specifying MRF's. [Cross and Jain 1983] applies a coding scheme to estimate the parameters in their binomial distribution models using a maximum likelihood criterion. [Elliott and Derin 1984] uses a least-square-fit method to estimate potential functions in the Gibbs distributions of their texture models. These methods are good when many uncorrupted realizations are available. When such data are difficult to acquire, choosing the clique potentials on an *ad hoc* basis has been reported to produce promising results [Geman and Geman 1984] [Marroquin, Mitter, and Poggio 1985]. Our experiments (Section 6) have also shown good results. These results are not surprising since the notion of clique potentials provides a simple mapping from "qualitative" spatial knowledge to numeric values of the parameters specifying the MRF's.

### 3. Bayesian Decision Rationale and Optimality of Solutions

At various levels of a visual hierarchy, estimations (decisions) must be made based on the information available. The estimation procedures become complex when the information is uncertain, which is usually the case in visual processing. In this section, we examine the Bayesian decision rationale and the optimal solutions to the labeling problem with respect to this rationale.

#### 3.1. *A Posteriori* Probabilities

Section 2 described how to encode prior spatial knowledge using the MRF formalism. Incorporating the image observations, Bayes' rule can then be used to derive the *a posteriori* probabilities on  $\Omega$  from the *a priori* model of the image.

$$\text{(Bayes' Rule)} \quad P(\omega | O) = \frac{P(\omega)P(O | \omega)}{\sum_{\omega' \in \Omega} P(\omega')P(O | \omega')} \quad (3.1)$$

$O$  denotes the image observations. The *likelihood* of an event  $\{X = \omega\}$  given  $O$ ,  $P(O | \omega)$ , is usually derived from the image degradation model involving imaging noise and blur [Geman and Geman 1984]. [Sher 1987] and [Bolles 1977] show methods to generate likelihood functions from either probabilistic models or statistical data.

For a shift-invariant point-spread function and white Gaussian noise, the *a posteriori* distribution associated with the *a priori* distribution defined by (2.3) and (2.4) is a Gibbs distribution with respect to a neighborhood system related to  $\Gamma$  and the support of the point-spread function [Geman and Geman 1984]. For simplicity, we assume the following conditional independence, that is generally true when the noise field is independently distributed.

$$P(O | \omega) = \prod_{s \in S} P(O_s | \omega_s) \quad (3.2)$$

$O_s$  denotes a set of image observations over a spatial region dependent on  $s$ , typically including  $s$  and its spatially adjacent elements. This assumption appears to be very useful for fusing and modeling early visual modules [Chou and Brown 1987a] and for texture modeling [Derin and Cole 1986]. The *a posteriori* MRF thus has the same neighborhood

system  $\Gamma$  with the energy function

$$U_O(\omega) = \sum_{c \in C} V_c(\omega) - T \sum_{s \in S} \ln P(O_s | \omega_s) \quad (3.3)$$

### 3.2. Optimal Labelings

The goodness of a labeling  $\hat{\omega}$ , following the Bayesian formalism, is evaluated in terms of its *a posteriori* expected loss,

$$Loss(\hat{\omega} | O) = \sum_{\omega \in \Omega} loss(\hat{\omega}, \omega) P(\omega | O) \quad (3.4)$$

where  $loss(\hat{\omega}, \omega)$  is a penalty associated with the estimate  $\hat{\omega}$  while the "truth" is  $\omega$ .

One question concerning the applicability of (3.4) is which loss function should be used for a given task. Except for few simple cases, the answer to this question usually relies on subjective judgements. One popular choice is assigning a constant penalty to incorrect estimates:  $loss(\hat{\omega}, \omega)$  equals to a constant (positive) value whenever  $\hat{\omega} \neq \omega$ , and 0 otherwise. Using this loss function, the configuration minimizing (3.4) maximizes the *a posteriori* probability  $P(\omega | O)$ , and therefore minimizes the *a posteriori* energy (3.3) in the MRF formalism. This Maximum A Posteriori (MAP) criterion has been widely applied to the labeling problem [Feldman and Yakimovsky 1974] [Geman and Geman 1984] [Derin and Cole 1986] [Murray and Buxton 1987] [Cohen and Cooper 1987]. Marroquin et al [1985] suggest that the number of mislabeled image entities of an estimation is a better loss measure for the labeling problem. They derive the Maximizer of the *a Posteriori* Marginals (MPM) estimation - choosing the configuration  $\hat{\omega} = (\hat{\omega}_{s_1}, \dots, \hat{\omega}_{s_N})$  such that

$$\hat{\omega}_s = \max_{l \in L} P_s(l | O) \quad \forall s \in S, \quad (3.5)$$

where  $P_s(l | O)$  denotes the *a posteriori* marginal probability of  $l$  on  $s$ . In their experiments the MPM estimator is shown to be superior to the MAP criterion when the signal to noise ratio is low.

Notice that the rationale of minimizing the loss function in (3.4) does not take the cost of computation into account, despite the fact that computational cost is usually a primary consideration in image understanding applications because of their immense configuration spaces. A sub-optimal estimator with an effective computation procedure would be much more useful than an optimal estimator that no one could ever compute. It is believed that the exact evaluation of MRF statistical moments, and therefore (3.4), is generally impossible since no analytic solutions exist [Hassner and Slansky 1980] [Geman and Geman 1984]. MAP and MPM can not be exactly determined for the same reason, except for some simple energy functions. In the rest of the paper, we discuss several numerical approaches for the approximate evaluations of the MAP and MPM estimations in the MRF formalism.

### 4. Stochastic Relaxation Methods

One method that has been successfully used to analyze the behavior of complex systems is generating sample configurations of a given system through stochastic simulations. Briefly, the Monte Carlo method of estimating the ensemble average of a variable  $Y(\omega)$ ,

$$\langle Y \rangle = \int_{\Omega} Y(\omega) dP(\omega),$$

is averaging its values over a set of samples  $\{\omega_1, \dots, \omega_R\}$  drawn from  $\Omega$ . If the sampling of  $\omega$ 's follows the distribution  $P$ , then  $\langle Y \rangle$  can be approximated by

$$\langle Y \rangle \approx \frac{1}{R} \sum_{r=1}^R Y(\omega_r).$$

We are interested in sampling procedures that generate configurations according to Gibbs distributions in the form of (2.3). With such procedures, the sample frequencies of the realizations of  $X$ , can be used as approximations for the marginal probabilities, i.e., MPM can be estimated; the configurations with higher probabilities are more likely to be sampled, and therefore MAP estimation becomes possible (see Section 4.2). Several procedures exist for this purpose. The basic idea of these procedures is to construct a regular Markov chain whose states correspond to the configurations of the system with the limiting distribution being the desired Gibbs distribution. That is, construct  $P_C$  - the transition matrix of the chain - in such a way that the following condition holds.

$$\pi P_C = \pi, \quad (4.1)$$

where  $\pi$  is the desired Gibbs measure. At equilibrium, the system's configurations are distributed according to  $\pi$  since  $\pi$  is the unique invariant measure of the constructed Markov chain [Kemeny and Snell 1960].

Consider each state transition of the Markov chain involving only the change of the state of a single entity in the system. To fulfill the requirement of the chain being regular, the procedure must continue to "visit" every entity. Let  $s(t)$  be the entity being visited at time  $t$ . The change of  $X_{s(t)}$  would result a change of the system energy by the amount specified by the configurations of those cliques that contain  $s(t)$  according to (2.4). Stochastic sampling procedures reminiscent of "relaxation" can be designed in the sense that the state transition of the entity being visited is stochastically decided by the states of the neighboring entities and itself. We will describe two of the *stochastic relaxation* procedures, namely the Metropolis algorithm [Metropolis et al. 1953] and the Gibbs sampler [Geman and Geman 1984], for their representativeness. Other variations basically follow the same principle and serve special purposes [Hassner and Slansky 1980] [Cross and Jain 1983] [Hinton and Sejnowski 1983].

#### 4.1. The Metropolis Algorithm and the Gibbs Sampler

Let  $X(t)$  denotes the state of the system at time step  $t$ . The state transition from step  $t$  to  $t+1$  of the Markov chain generated by the Metropolis sampling algorithm consists of two basic steps:

- (1) Randomly select a new configuration  $\omega'$  (randomly visit an entity  $s$  and choose a new state  $\omega'_s$ ), and compute the energy change  $\Delta E = E(\omega') - E(X(t))$ .
- (2) If  $\Delta E < 0$ , set  $X(t+1) = \omega'$ . Otherwise, set  $X(t+1)$  to  $\omega'$  or  $X(t)$  with probabilities  $\frac{\pi(\omega')}{\pi(X(t))} = e^{-\Delta E/T}$  and  $1 - e^{-\Delta E/T}$  respectively.

Allowing transitions with energy increases, a common characteristic of all stochastic relaxation procedures, prevents the sampling process from getting stuck at states of local

energy minimum - an undesirable property of every deterministic hill-climbing procedure. In contrast to the explicit use of the energy difference in the Metropolis algorithm, the Gibbs sampler uses the local characteristics to construct a Markov chain. A state transition of the Gibbs sampler also consists two steps:

- (1) Visit an entity  $s$ .
- (2) Randomly select the new state  $\omega'_s$  for  $X_s(t+1)$  following the distribution  $\pi(X_s(t+1)=\omega'_s | X_r(t), r \neq s)$ . Having the form in (2.5), this distribution is generally easy to compute.

For binary systems, the Gibbs sampler is equivalent to the widely used "Heat Bath" algorithm - changing the state with probability  $\frac{1}{1+e^{\Delta U/T}}$ . Like other relaxation methods, the above procedures suggest the use of a parallel implementation since "updating" the  $X_s$ 's requires propagating information only among neighboring computing units. Extra caution must be paid to the updating patterns of synchronous machines. For the Metropolis and Heat Bath algorithms, using any prescribed updating order may result in the Markov chain not converging to the desired Gibbs distribution  $\pi$  [Marroquin 1985]. Our experiments use the Gibbs sampler exclusively because it guarantees the coincidence of  $\pi$  with the invariant measure of the chain as long as neighboring entities are not updated simultaneously.

#### 4.2. The Monte Carlo and Simulated Annealing Methods

The stochastic relaxation scheme can be used to approximate the *a posteriori* marginal probabilities for the MPM estimation by simulating the equilibrium behavior of the *a posteriori* MRF. Since the Markov chain constructed by either the Metropolis algorithm or the Gibbs sampler leads to the desired limiting distribution regardless of its initial state, the law of large numbers suggests the marginal probability  $P_s(l|O)$  be approximated by the sample frequency of  $X_s=l$  at equilibrium, that is,

$$P_s(l|O) \approx \frac{1}{n-k} \sum_{t=k}^n \delta(X_s(t)-l) \quad (4.2)$$

where  $\delta(0) = 1$ , and 0 elsewhere.  $k$  is the number of steps for the chain to reach equilibrium, and  $n$  is the total number of steps of the simulation. Practically, experimentation is needed to determine how large  $n$  and  $k$  should be to achieve a desirable approximation accuracy given an arbitrary MRF. Cross and Jain [1983] have observed that in less than 10 iterations (full sweeps over the image entities), their texture modeling system becomes "stable" when sampled by a variation of the Metropolis algorithm. In general, in the order of hundreds of iterations are needed for the MPM estimation.

The system temperature -  $T$  in (2.3) - also plays an important role in MRF simulations. With low temperatures, the Gibbs distribution strongly favors the low energy configurations, but the time required for the system to reach equilibrium may be long. The system may reach equilibrium faster at higher temperatures, but the configurations are more evenly sampled; i.e., it may require more samples to make accurate MPM estimations. The idea of *simulated annealing* [Kirkpatrick, Gelatt, and Vecchi 1983], obviously inspired by physical annealing, is to reach the minimum energy states of a system by starting the system at a high temperature and gradually reducing it. In doing so the system tends to respond to large energy differences at the beginning, and is likely to find a good minimum

energy state independent of its starting state. As the temperature decreases, the system tends to respond to small energy differences, and ideally settles at the lowest energy states ever encountered. The decreasing sequence of temperatures, called the annealing schedule, decides the effectiveness of this process. If the time spent at each temperature is not enough, the system may not converge to the global minimum states. On the other hand, it is often computationally prohibitive to use a slowly decreasing schedule. Geman and Geman [84] have derived an upper bound for the annealing schedules so that the schedules slower than this bound are guaranteed to converge to the global minimum energy states. However, this bound is very difficult to decide in practice since it relates to the range of energy values of the system.

Simulated annealing has been applied in many computer vision tasks that involve optimization over exponential spaces, including the MAP estimation [Geman and Geman 1984] and the stereo matching problem [Barnard 1987]. One major concern of using the stochastic relaxation scheme is its efficiency: at what cost can this scheme deliver satisfactory results? Not surprisingly, the cost is intolerable for many applications. In the next section, we describe a new deterministic method to approximate MAP. This method, following a search path suggested by the visual observations to find a minimum energy state, appears to give results favorably comparable in practice to the existing relaxation methods while being computationally less expensive.

## 5. Deterministic Relaxation Methods

Exact calculation of the MAP estimate is computationally prohibitive. For vision systems that require predictable results in reasonable time periods, using suboptimal estimation criteria and/or heuristics in searching for solutions seems to be a reasonable alternative to the stochastic relaxation scheme. In [Derin and Cole 1986], MAP estimations are performed on narrow strips of the image. The strips are limited to at most four rows wide so that MAP can be exactly computed for each strip by a dynamic programming algorithm at feasible cost. For each estimation, only the estimate of the first row of a strip is kept. It serves as the boundary condition for the next strip consisting of the rest of the rows and a new one. Though limiting the extent of the (column-wise) interactions, the texture segmentation results appear to be impressive. Before we describe the proposed heuristic-based algorithm, we examine an iterative relaxation method for estimating MAP.

### 5.1. Iterative Energy Minimization

A simple version of deterministic iterative relaxation methods for energy minimization is the Metropolis algorithm without randomness: Start with an initial configuration. At each iteration through the image entities, the state of each entity is either changed to the state that yields maximal decrease of the energy, or is left unchanged if no energy reduction is possible. The process stops when no more changes can be made. This algorithm is guaranteed to find a local minimum of the energy function since each iteration strictly decreases the energy value and there are only a finite number of different values of the energy function. For parallel implementation, convergence is assured if the neighboring entities are not updated simultaneously.

Unavoidably, the local minimum obtained by the above algorithm may be far from optimal. Two enhancements are apparently helpful:

- (1) Start with a better initialization of the MRF. The best one can hope is that the energy value of the initial configuration falls into the valley of the global minimum. One possibility is to use the *maximum likelihood estimates* (MLE) -  $X_s(0) = \omega_s$ , if  $\max_{l \in L} P(O_s | l) = P(O_s | \omega_s)$ .
- (2) Escape from shallow valleys. By changing the states of more than one entity at once, the new configuration may lead to a better local minimum. In a procedure described in [Cohen and Cooper 1987], the entities with small preferences of the current states over the others are assigned new states when a local minimum is reached. The relaxation restarts with the new configuration as the initialization. At each convergence, the magnitude of the local minimum is estimated, The procedure halting when no significant change of the magnitudes is observed. The hope is that the deepest valley will be found in this process.

Unfortunately, these two modifications are not adequate. The local MLE's are good only when the noise process is correctly modeled in computing the likelihoods and there are significant differences among the likelihoods of the hypotheses. Frequently these conditions cannot be met. Cohen and Cooper's procedure, obviously a compromise between stochastic and deterministic relaxation methods, suggests a tradeoff between speed and performance.

The algorithm of this paper blends the initialization into the estimation process. Instead of stepping through the configuration space  $\Omega$ , this algorithm constructs a configuration with a local minimal energy measure. Observable evidence and spatial prior knowledge are combined in the process of the construction, resulting in better results and efficiency. The details of this algorithm are described next.

## 5.2. The Highest Confidence First Algorithm

To see how this algorithm works, some terminology needs to be introduced. Let  $\bar{L} = L \cup \{l_0\}$  denote the *augmented label set*, where  $L = \{l_1, \dots, l_Q\}$  is the set of labels, and  $l_0$  is the null label corresponding to the "uncommitted" state in the construction. Let  $\bar{\Omega} = \{\omega = (\omega_1, \dots, \omega_N) | \omega_s \in \bar{L}, \forall s \in S\}$  denote the *augmented configuration space*. Define the *augmented a posteriori local energy* of  $l \in L$  with respect to  $s \in S$  and a configuration  $\omega \in \bar{\Omega}$  as

$$E_s(l) = \sum_{c: s \in c} V'_c(\omega') - T \ln P(O_s | l), \quad (5.1)$$

where  $\omega' \in \bar{\Omega}$  is the configuration that agrees with  $\omega$  everywhere except  $\omega'_s = l$ , and  $V'_c$  is 0 if  $\omega_r = l_0$  for any  $r$  in  $c$ , otherwise it is equal to  $V_c$  - the potential function.

The basic idea of this algorithm is to construct a sequence of configurations  $\omega^0, \omega^1, \dots$  with the starting configuration  $\omega^0 = (l_0, \dots, l_0)$ , and a terminal configuration  $\omega^f \in \bar{\Omega}$ , where  $U_O(\omega^f)$  is a local minimum with respect to  $\bar{\Omega}$ . We say an entity  $s$  has *made its current decision*  $l$ ,  $l \in L$ , if the just-constructed (current) configuration  $\omega$  in the sequence has the component  $\omega_s = l$ , and it has not made a decision if  $\omega_s = l_0$ . Once an entity makes a decision, it can *change* this decision to other labels of  $L$  but not to  $l_0$ . To ensure the quality of the resulting estimate -  $\omega^f$ , at each step of the construction we permit only the least "stable" entity to change/make its decision. We define the *stability* of  $s$  with respect to the current configuration  $\omega$ ,  $\omega_s = l$ , as

$$G_s(\omega) = \min_{k \in L, k \neq l} \Delta E_s(k, l) \quad \text{if } l \in L \quad (5.2a)$$

$$G_s(\omega) = - \min_{k \in L, k \neq j} \Delta E_s(k, j) \quad \text{if } l = l_0 \text{ and } j \in L \text{ s.t. } E_s(j) = \min_{k \in L} E_s(k), \quad (5.2b)$$

where  $\Delta E_s(j, k) = E_s(j) - E_s(k)$  with respect to  $\omega$ .

The stability defined above is a combined measure of the observable evidence and the *a priori* knowledge about the preferences of the current state over the other alternatives. A negative value of  $G$ , that is always true for (5.2b), indicates a more stable configuration will result from an alternative decision. Since an entity has no effect on its neighbors unless it has made a decision, the entities with large likelihood ratios of one label over the others - strong external evidence in favor of a label - will be visited early in the construction sequence. The entities with little idea from the observations will collect information from the neighbors' decisions to make their decisions; an early decision will be altered if the neighbors' later decisions are strongly against it. In this way, every step of this construction makes a maximal progress based on the current knowledge about the field - the  $G$ 's. This *Highest Confidence First* algorithm is expected to find the estimate that is "consistent" with the observations and the *a priori* knowledge.

The Highest Confidence First algorithm can be implemented serially with a heap (priority queue) maintaining the visiting order of the construction according to the values of  $G$ 's in such a way that the *top* of the heap is the entity with the smallest  $G$  value. Updating the *top*'s decision will cause the changes of its neighbors'  $G$ -values, and therefore the structure of the heap. The following is the pseudo code for the Highest Confidence First algorithm:

```

 $\omega = (l_0, \dots, l_0);$ 
 $top = \text{Create\_Heap}(\omega);$ 
while ( $G_{top} < 0$ ) {
     $s = top;$ 
    Change_State( $\omega_s$ );
    Update_G( $G_s$ );
    Adjust_Heap( $s$ );
    foreach ( $r \in N_s$ ) {
        Update_G( $G_r$ );
        Adjust_Heap( $r$ );
    }
}
return( $\omega$ );

```

Change\_State( $\omega_s$ ) changes the current state  $\omega_s$  of  $s$  to the state  $l$  such that  $\Delta E_s(l, \omega_s) = \min_{k \in L, k \neq \omega_s} \Delta E_s(k, \omega_s)$  if  $\omega_s \in L$ , or  $E_s(l) = \min_{k \in L} E_s(k)$  if  $\omega_s = l_0$ . Upon this change taking place, the stability of  $s$  changes to positive. Update\_G is called for every entity that is affected by this change, namely the neighbors of  $s$  according to (5.1), to update their stability measures with respect to the new configuration. Adjust\_Heap( $r$ ) maintains the heap property by moving  $r$  up or down according to its updated  $G$ -value.

Several desirable properties of this procedure can easily be verified:

- (1) **Termination:** This procedure always returns in finite time. To see this property, let us consider the two types of *Change\_State* - making and changing a decision - separately. The procedure can make at most  $N$  decisions, one for each entity, since nullifying decisions is impossible. Let  $D = (S_D, S - S_D)$  be a *partition* of  $S$  such that  $S_D$  is the set of entities that have made decisions. Let  $\bar{\Omega}_D = \{\omega \in \bar{\Omega} \mid \omega_s \in L \ \forall s \in S_D, \text{ and } \omega_s = l_0 \ \forall s \in S - S_D\}$ . Since, by (5.1) and (5.2a), changing the decision of  $s \in S_D$  strictly decreases the function  $U_D : \bar{\Omega}_D \rightarrow R$ ,

$$U_D(\omega) = \sum_c V'_c(\omega) - T \sum_{s \in S_D} \ln P(O_s \mid \omega_s),$$

the procedure can make only a finite number of changes with respect to a fixed partition  $D$ . There are only a finite number of partitions, therefore the total number of decision changes is finite.

- (2) **Feasibility:** The returned configuration is in  $\Omega$  - the space of feasible solutions. For if otherwise, there exists an  $s$  such that  $\omega_s = l_0$ . From (5.2b),  $G_s < 0$ . This violates the heap invariant property since it requires  $G_{top} \geq 0$  to exit the while loop.
- (3) **Optimality:** The returned configuration has the locally minimal energy measure with respect to  $\Omega$ . That is, changing the decision of any single entity can not decrease the *a posteriori* energy measure  $U_O$ . As above, this property can easily be derived from (5.2a) and the heap properties.

This implementation takes  $O(N)$  comparisons to create the heap and  $O(\log(N))$  to maintain the heap invariance for every visit to an entity, provided the neighborhood size is small relative to  $N$ . The overheads of heap maintenance are well repaid since the procedure makes progress for every visit, in contrast to the iterative relaxation procedure (Section 5.1) that may make only few changes per iteration ( $N$  visits). Our experiments show that on the average, less than one percent of the entities are visited more than once using the proposed algorithm while the deterministic relaxation procedure takes around 10 iterations to reach a local minimum. This advantage becomes more evident as the number of entities gets larger. Experimental results are strongly in favor of the proposed algorithm for both efficiency and correctness. They are discussed in Section 6.

### 5.3. Possible Extensions

Since the order of the deterministic decisions of the entities in a cooperative network is crucial to the final mutual agreement, the proposed algorithm assumes that good results can be obtained by delaying the decisions of those entities who have little idea about what to do until they get enough help from their neighbors. This heuristic can be used along with other computational methods to achieve, perhaps, better results.

Let us look more closely at the process of achieving a consensus using this heuristic. At each stage,  $S_D$  consists of a set of isolated clusters. A cluster is a set of spatially connected (with respect to  $\Gamma$ ) entities. We say two clusters are isolated from each other if none of the entities of a cluster is a neighbor of any entity of the other cluster. Each cluster corresponds to an MRF with free boundaries in our formalism. When an entity makes a decision, a cluster is created or expanded, or clusters are merged. When an entity changes a decision, the energy of the corresponding MRF is reduced. Eventually, all the clusters are merged and the final agreement corresponds to a local minimum configuration of the

corresponding MRF.

The notion of growing clusters suggests a natural partition of the image. At any instance, the entities belonging to the same cluster are tightly related, but they are independent of the members of other clusters. The addition of a new member to a cluster may change the decisions of the old members, but the changes are expected to be small due to the way the clusters are constructed. Therefore, it makes sense to compute the MAP estimates exactly for small clusters early in the process. We believe that by doing so the results would be better than the results using the horizontal strip partition as in [Derin and Cole 1986].

The process of growing clusters is similar to annealing in the sense that it responds to large energy differences earlier than small ones. Nondeterminism can be introduced to those entities that stay "unstable" - the entities on or exterior to the border of the clusters - late in the process, since more spatial information is required for them to reach a globally satisfactory agreement.

The Highest Confidence First algorithm can be implemented with a set of cooperative computing units. Consider a *winner-take-all* network where each unit corresponds to an entity of the image [Feldman and Ballard 1981]. Only the units with the smallest stability measures can "fire" at one instant; each unit maintains the knowledge about the neighboring units so that its stability measure can be updated immediately should any neighbor change its state. The parallelism gained, however, is limited due to the sequential firing order.

## 6. Experiments and Results

We have chosen to tackle the well studied problem of edge detection using MRFs as the underlying formalism. The labeling problem in this context is to assign to each edge element a label from the set {EDGE, NON-EDGE}. Each of these edge elements is modeled as an MRF entity. The MRF entities are considered to be situated on the boundary between two pixels (see Fig. 1). The MRF model used is similar to the "Line Process" MRF used both by Geman *et al* [1984] and Marroquin *et al* [1985]. Hence the MRF is binary, with  $2(N^2 - N)$  entities where the image is a  $N \times N$  rectangular pixel array.

### 6.1. Construction of Potential Functions to Encode Prior Knowledge

The spatial relationships between entities we wish to enforce include:

- (1) To encourage the growth of continuous line segments,
- (2) To discourage abrupt breaks in line segments,
- (3) To discourage close parallel lines (competitions) and
- (4) To discourage sharp turns in line segments.

A second order neighborhood turns out to be sufficient to enforce all the relationships we want. In this neighborhood system, each MRF element is adjacent to eight others (see Figs 1 and 2).

The second order neighborhood has cliques of sizes 1 through 4 (see Fig. 3). The potential values we assign to various configurations of these cliques are shown in Fig. 4. These values form the specification of the potential functions. Therefore potential functions can be

seen to be specified by about 10 parameters, which are currently assigned in an *ad hoc* manner. The rules of thumb that are used to assign values to these parameters are:

**Determine Structure Enforcers** For each clique, attempt to determine what kind of structural relation it is uniquely capable of enforcing.

**Encode Prior Structural Knowledge** By assigning "high" potential values to undesirable configurations of the cliques and "low" values to desirable ones, we attempt to ensure that the final estimate will contain as few of the undesirable ones as possible.

**Encode Statistical Prior Knowledge** We use the clique consisting of the singleton node to bring the first order statistics (e.g. the density of EDGES) of the MRF into line with what we already know. The potential of the clique when the MRF entity is an EDGE is set to our estimate of the log of the (local) odds of an entity being an EDGE over a NON-EDGE, and is set to 0 when it is a NON-EDGE.

A point to be noted is that some of these parameter values are interdependent. For example, increasing the energy for "break" (Fig. 4b) and "continuation" (Fig. 4c) configurations simultaneously would be of little use, as the increases would tend to cancel each other out.

The sensitivity of the results obtained to changes in the parameters specifying the potential functions depends upon the parameter in question. Our experience is that changing the potential function associated with the 1-clique had the greatest effect on the final result, followed by the 2-clique and 4-clique potential functions, in that order. This could be because the singleton clique controls first order statistics and the larger cliques higher order statistics, which are known to be less important in distinguishing images [Julesz 1981].

## 6.2. Likelihood Generation

We adopt a step-edge with white Gaussian noise model to compute the local likelihoods of an entity  $s$  being EDGE or NON-EDGE -  $P(O_s | \omega_s = \text{EDGE})$  and  $P(O_s | \omega_s = \text{NON-EDGE})$ . The observation -  $O_s$  - is a  $1 \times 4$  or  $4 \times 1$  window of brightness observations surrounding  $s$ . This window of intensity values is assumed to be a realization of one of the possible events depicted in Figure 5, corrupted by independent Gaussian noise. The reader is referred to [Sher 1987] for details of probabilistic edge detection.

From (3.2), observe that scaling  $P(O_s | l)$  for every  $l \in L$  by a constant factor for fixed  $s$  does not change the *a posteriori* distribution. This fact allows us to use the likelihood ratios -  $\frac{P(O_s | \omega_s = \text{EDGE})}{P(O_s | \omega_s = \text{NON-EDGE})}$  - as the only input data, thus simplifying the computation of the stability measures (5.2). Thresholding the likelihood ratios by the prior (local) odds of an entity being an EDGE results in the thresholded likelihood ratio (TLR) configuration that can be considered as an MAP estimate obtained without using contextual information. In our experiments, we use TLR as the initial configuration whenever possible.

## 6.3. A General Purpose MRF Simulator

Our experiments use an interactive general-purpose MRF simulator package with extensive graphics and menu-driven control (Fig. 6). This package takes the description of the MRF and the likelihood ratios as input and simulates the state transitions of the

entities comprising the the MRF. The user can specify the estimation algorithm to be used and also the initialization of the MRF - each entity can be initially set to either a NON-EDGE or to its TLR state. The input MRF is constrained to be a homogeneous one, so as to make the time and space needed to run simulations reasonable.

The user provides the description of the MRF to the simulator in a file. This file contains a specification of the nodes comprising each clique, as well as the potential function associated with it. The user specifies all cliques that a node could belong to in the most general case, including all instances of a particular clique type that contain the node. (i.e., even if all the cliques containing a node are instances of the same clique type, the user specifies each instance separately). The nodes forming a clique are specified by their coordinates relative to the node of interest, which is defined to be at relative coordinates (0, 0). Boundary conditions, as in the case of nodes near the border of the MRF, are taken care of by the simulator. The potential function is specified as a function that takes as input a configuration vector (a vector of states of nodes of the MRF) and returns a potential value. The potential function is associated with the clique description, and the ordering of the node states in the configuration vector passed it is the same as the order the nodes are specified in the description of the clique itself.

The simulator performs certain preprocessing actions on the description of the MRF provided by the user, to promote run-time efficiency. The first is to store each potential function as a table indexed into by a configuration vector. This is done so as to avoid run-time calling of the user's potential function code, which can be quite complex, replacing it instead with a simple table lookup. The other is "clique containment", which is based on the observation that if one clique completely contains the other, then a configuration vector of the nodes in the larger clique contains implicitly the configuration vector for the smaller clique. This suggests that by judiciously "adding" together the potential functions for the clique in the preprocessing stage, we can avoid run-time evaluation of the potential function for the smaller clique. This simplifies the state transition energy evaluation by reducing the number of terms to be summed up. If floating-point arithmetic is costly, this can save considerable computational effort. The preprocessing needs to be done just once, and can be performed off-line.

#### 6.4. Experimental Results

The simulator described above has been used for a series of experiments aimed at comparing the performances of various relaxation algorithms with respect to the goodness of final estimations and rate of convergence. We focus upon algorithms using the MAP criterion, including Highest Confidence First (HCF), Deterministic Iterative Relaxation (DIR) and stochastic MAP (simulated annealing with Gibbs Sampler [Geman and Geman 1984]). The results obtained by using stochastic MPM (Monte Carlo approximation to the MPM estimate [Marroquin, Mitter, and Poggio 1985]) are also presented for the sake of completeness of comparisons, as are those obtained by applying  $3 \times 3$  Kirach operators with non-maximum suppression. The annealing schedule for the stochastic MAP follows the one suggested in [Geman and Geman 1984], i.e.  $T_k = \frac{c}{\log(1+k)}$  where  $T_k$  is the temperature for the  $k^{th}$  iteration, with  $c = 4.0$ . The stochastic MAP was run for 1000 iterations and the stochastic MPM for 500 (300 to reach equilibrium, 200 to collect statistics).

#### 6.4.1. Comparison of Estimates

Here we show the results of three sets of experiments (Figs 7 through 9). The figures for each set contain the original image, the result from the Kirsch operators, the TLR configuration and the results obtained by using stochastic MAP, stochastic MPM, DIR (scan-line visiting order), DIR (random visiting order) and HCF algorithms. Except in the case of the HCF algorithm, where the MRF is initialized to all null (uncommitted) states, the MRF is initialized to the TLR configuration. The MRF specification is the same throughout.

Fig. 7a shows a synthetic 50 pixel square "checkerboard" pattern. Each of patches is 10 pixels across, with an intensity chosen randomly from between 0 and 255. The image has been degraded by independently adding to each pixel Gaussian noise with a mean of 0 and a standard deviation of 16. The HCF and stochastic MPM (Figs. 7g and 7d) are the same, and have completed most of the desired edges. The DIRs (Figs. 7d and 7e) have incomplete edges and the stochastic MAP has some undesired edges and incomplete desired edges (Fig. 7c). The Kirsch operator result is not shown as the edges in this image are always located exactly in between pixels, while the Kirsch operator assumes edges to be at pixel locations, and so a comparison would be unfair to the Kirsch operators.

Fig. 8a shows a 50 pixel square natural image of a wooden block with the letter "P" on it. The MAP estimate has several undesirable lines (Fig. 8d). The MPM estimate performs poorly on the right edge of the block and the inner ring of the "P". The DIR scheme (serial scan) (Fig. 8f) performs better than the random scan version (Fig. 8g), but is less than satisfactory on the leg of the "P" and the right edge of the block. The HCF estimate (Fig. 8h) does not suffer from the above flaws, producing clean, connected edges.

Fig. 9a shows a  $100 \times 124$  natural image of 4 plastic blocks with the letters "U", "R", "C" and "S" on them. Again, the HCF algorithm produces superior results (Fig. 9g). It has the clearest letter outlines and also is alone in detecting the entire bottom edge of the "R" block. The MAP estimate partially detects the bottom edge of the "R" block, but generates redundant lines (Fig. 9c). The MPM estimate has clear letter outlines but does poorly on the outlines of the left blocks (Fig. 9d). The DIR scheme (scan-line) does well on the letter outlines but poorly on the block outlines while the random scan version does poorly on both (Figs. 9e and 9f).

To test the *robustness* of the algorithms, we conduct further experiments using a likelihood generator with a less complete edge model. Since offset edges (Fig. 5c) are not considered here, multiple responses become significant as can be seen from the TLR configuration shown in Fig. 10a. This change strongly affects the estimates produced by all the algorithms except the HCF, as can be seen from comparing corresponding pictures in Fig. 9 and Fig. 10.

#### 6.4.2. Rates of Convergence

We restrict ourselves to comparisons between deterministic schemes, as stochastic schemes do not have any convergence criterion *per se* - the point of convergence is dependent upon our judgement as to when equilibrium has been reached, and as to when we have gathered enough statistics to estimate the joint (or marginal) probabilities accurately (typically several hundred iterations are needed). The deterministic algorithms (HCF and

DIR (scan-line)) have been timed on images of various sizes using a Sun 3/260 with floating point acceleration. The results are shown in Table 1.

## 6.5. Analysis of Experimental Results

### *Goodness of Estimates*

- (1) The HCF algorithm repeatedly outperforms all other algorithms, giving superior results both with synthetic and real image data. The common characteristics of the results we have obtained from using this algorithm are that they all fit well in our model of the world, which consists of smoothly continuous boundaries, and that they are consistent with the observations.
- (2) The HCF algorithm also appears to be robust, in that it produces an estimate consistent with the observations even when the MRF model used is inadequate, as in the experiment using the less sophisticated edge detector. Since our MRF model does not take into account multiple responses, the MAP criterion may not lead to the "best" results. In this case, the local minimum found by the HCF algorithm is actually better than the global one as it is based on the strength of external evidence.
- (3) The DIR algorithm performs inconsistently and its results depend to a large extent upon the initialization of the MRF and the visiting order. It is also not clear which, if any, of the visiting orders studied is better than the other.
- (4) The stochastic MAP algorithm with simulated annealing gets stuck in undesirable local minima, suggesting that our annealing schedule might have lowered the temperature too fast. However, an appropriate annealing schedule seems hard to obtain *a priori*.

### *Convergence Times*

- (1) The HCF algorithm makes a perhaps surprisingly small number of visits before converging. Clearly, due to the initialization, it must visit every node at least once. What is surprising is that it visits each node on the *average* less than 1.01 times before converging. What this implies is that the first decision made by a node is nearly always the best one.
- (2) The convergence times of the DIR algorithms are unpredictable - they vary with visiting order, MRF initialization and even upon the particular image given as input. The HCF algorithm, in contrast, takes almost the same time on different images of the same size. The time taken by the HCF algorithm includes the time taken to set up the heap initially. This may, in some circumstances, be a little unfair. For instance, if one has to process data online from various information sources [Chou and Brown 1987a] [Poggio 1985], the heap setting up cost can be treated as a preprocessing cost rather than a run-time one.
- (3) In theory, the time taken by the HCF algorithm should be given by  $c_1N + c_2V\log_2N$ , where  $c_1$  and  $c_2$  are positive constants,  $N$  the number of entities to be labeled and  $V$  the number of visits.  $V$  here is at least  $N$  and we conjecture that on the average it is  $cN$  for some small ( $1 < c < 2$ ) constant  $c$ . Since the latter term should dominate, one would expect to see a nonlinear curve in a plot of run time *vs.* number of entities. However, the curve is very nearly a straight line. which indicates either that the

constant  $c_2$  is very small, or that the changed stability values do not propagate very far up the heap on the average. The former does not appear to be true, as our experiences suggest that the initial heap construction takes far less time than the rest of the algorithm.

## 7. Conclusions and Future Research

We have described a new approach for solving the labeling problem. The Highest Confidence First algorithm, aimed at approximating the MAP estimate with *a priori* knowledge modeled by MRF's and external observations represented as likelihoods, leads to outstanding results in our experiments with both synthetic and natural images. Not only is this algorithm much faster than stochastic estimation procedures, it also converges predictably. In addition, the algorithm is robust - in the case that the prior model proves inadequate, it produces an estimate that is highly consistent with the observations.

We are incorporating the Highest Confidence First algorithm in a multi-modal segmenter described in [Chou and Brown 1987a] and believe it to be well suited to a scenario where the result is to be computed incrementally from sparse and dynamically-arriving data, possibly from multiple sources.

We are studying methods for systematically specifying the clique potential functions of MRF's from given realizations. We are also analyzing the rapid convergence of the HCF algorithm observed in our experiments from a theoretical viewpoint.

The concept of a confidence-based heuristic is likely to be useful whenever there is a set of cooperating processes attempting to reach a consensus. The idea that the processes with a greater degree of certainty about their decision, get to make it first, is intuitively appealing. We are investigating applications of this idea to other fields.

## Acknowledgements

We thank Chris Brown for helpful discussions on this subject. Without his active encouragement and support this work would have been impossible.

We are indebted to Dave Sher for suggesting the low-level edge models used in this work, and for the use of edge-detection software developed by him for our experiments.

The first author would also like to express his gratitude to the members of his thesis committee (Dana Ballard, Chris Brown, Jerry Feldman and Henry Kyburg) for their valuable suggestions.

## References

- Barnard, S., "Stereo Matching By Hierarchical, Microcanonical Annealing", *Proceedings: Image Understanding Workshop 2* (Feb. 1987), 792-797.
- Besag, J., "Spatial interaction and the statistical analysis of lattice systems (with discussion)", *Journal of Royal Statistics Society, series B* 36 (1974), 192-326.
- Bolles, R. C., "Verification Vision for Programmable Assembly", *Proceedings: IJCAI-77*, Aug. 1977, 569-575.

Chou, P. B. and C. M. Brown, "Probabilistic Information Fusion for Multi-Modal Image Segmentation", *Proceedings: IJCAI-87*, Milan, Italy, Aug. 1987.

Chou, P. B. and C. M. Brown, "Multi-Modal Segmentation Using Markov Random Fields", *Proceedings: Image Understanding Workshop 2* (Feb. 1987), 663-670.

Cohen, F. S. and D. B. Cooper, "Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, 2 (Mar. 1987), 195-219.

Cross, G. R. and A. K. Jain, "Markov Random Field Texture Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5*, 1 (Jan. 1983), 25-39.

Derin, H. and W. S. Cole, "Segmentation of Textured Images Using Gibbs Random Fields", *Computer Vision, Graphics, and Image Processing 35* (1986), 72-98.

Duda, R. O., P. E. Hart, and N. J. Nilsson, "Subjective Bayesian Methods for Rule-Based Inference Systems", SRI Technical Note 124, SRI International, 1976.

Elliott, H. and H. Derin, "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields", #ECE-UMASS-SE84-1, University of Massachusetts, Sept. 1984.

Feldman, J. A. and Y. Yakimovsky, "Decision Theory and Artificial Intelligence: I. A Semantics-Based Region Analyzer", *Artificial Intelligence 5* (1974), 349-371.

Feldman, J. A. and D. H. Ballard, "Computing with connections", TR 72, Computer Science Department, Univ. of Rochester, 1981.

Geman, S. and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*, 6 (Nov. 1984), 721-741.

Hassner, M. and J. Slansky, "The Use of Markov Random Fields as Models of Texture", in *Image Modeling*, Rosenfeld, A. (editor), Academic Press, Inc., 1980, 185-198.

Hinton, G. E. and T. J. Sejnowski, "Optimal Perceptual Inference", *Proceedings: IEEE Conf. CVPR*, 1983, 448-453.

Julesz, B., "Textons, the elements of texture perception, and their interactions", *Natural 290 12* (March, 1981), 91 - 97.

Kanal, L. N., "Markov Mesh Models", in *Image Modeling*, Rosenfeld, A. (editor), Academic Press, Inc., 1980, 239-243.

Kemeny, J. G. and J. L. Snell, *Finite Markov Chains*, Van Nostrand, New York, 1960.

Kindermann, R. and J. L. Snell, "Markov Random Fields and their Applications", in *Contemporary Mathematics*, vol. 1, American Mathematical Society, 1980.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing", *Science* 220 (1983), 671-680.

Marroquin, J., S. Mitter, and T. Poggio, "Probabilistic Solution of Ill-Posed Problems in Computational Vision", *Proceedings: Image Understanding Workshop*, Dec. 1985, 293-309.

Marroquin, J. L., "Probabilistic Solution of Inverse Problems", AI-TR 860, MIT Artificial Intelligence Laboratory, Sep. 1985.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines", *Journal of Chemical Physics* 21 (1953), 1087-1091.

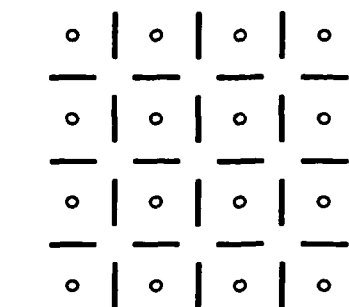
Murray, D. W. and B. F. Buxton, "Scene Segmentation from Visual Motion Using Global Optimization", *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, 2 (Mar. 1987), 220-228.

Pearl, J., "Fusion, Propagation, and Structuring in Bayesian Networks", Computer Science Dpt.-850022 R-42, Apr. 1985. Computer Science Dept., UCLA.

Peleg, S., "A new probabilistic relaxation scheme", *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2* (1980), 362.

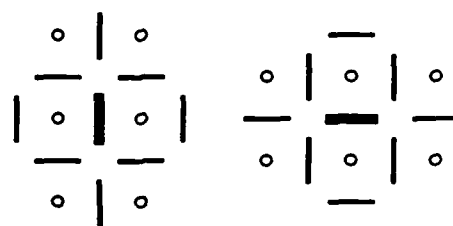
Poggio, T., "Integrating vision modules with coupled MRFs", Working Paper No. 285, MIT A.I. Lab., Dec. 1985.

Sher, D. B., "Advanced Likelihood Generators for Boundary Detection", TR197, Univ. of Rochester, Computer Science Dpt., Jan. 1987.



— : Line (MRF) Element  
 ○ : Pixel (Image)

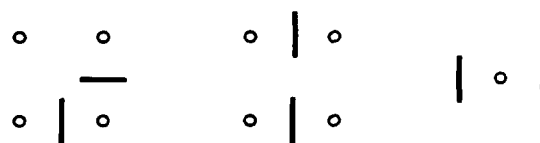
Figure 1



(a)

(b)

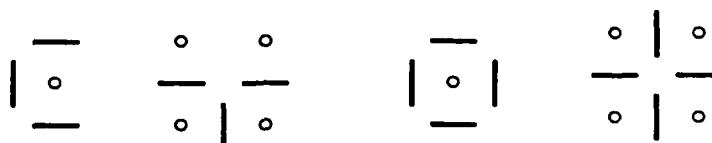
Figure 2



(a)

(b)

(c)



(d)

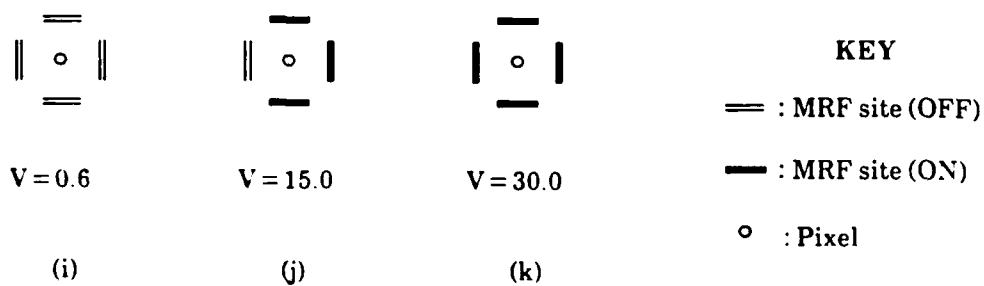
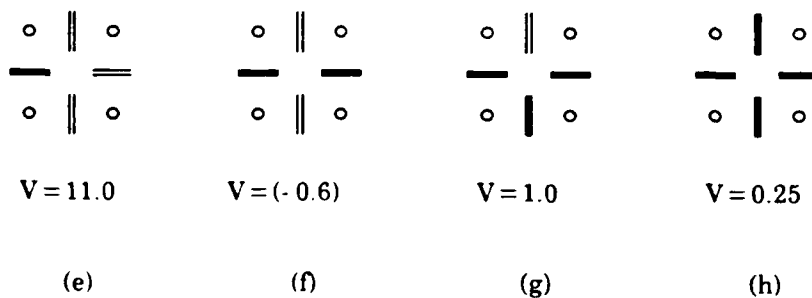
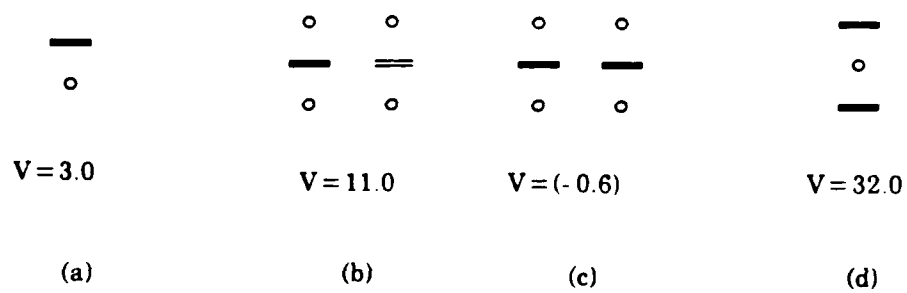
(e)

(f)

(g)

Figure 3

Figure 1: Relationship between MRF entities and pixels. Figure 2: The second-order neighborhood system. Figure 3: Cliques in neighborhood system, of size greater than one. (a)-(c): size 2; (d)-(e): size 3; (f)-(g): size 4



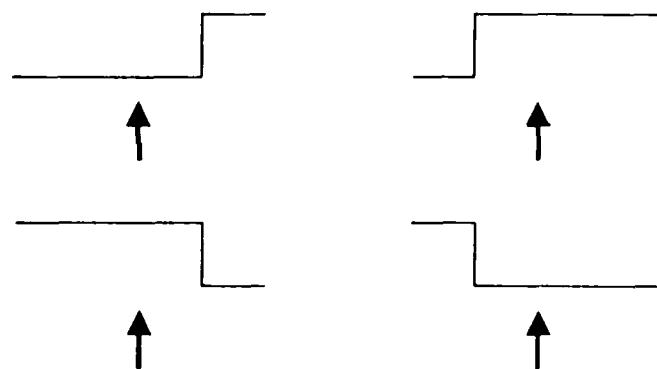
**Figure 4: Potential Assignments for Cliques**  
 (Configurations not shown have null (0.0) potential values)



(a)



(b)



(c)

**Figure 5:** Image events in a 4X1 window  
 (a) Edge occurring at center of window,  
 (b) Homogenous region: no edge occurs,  
 (c) No edge at center, offset edge occurs.  
 (Arrow indicates center of window)

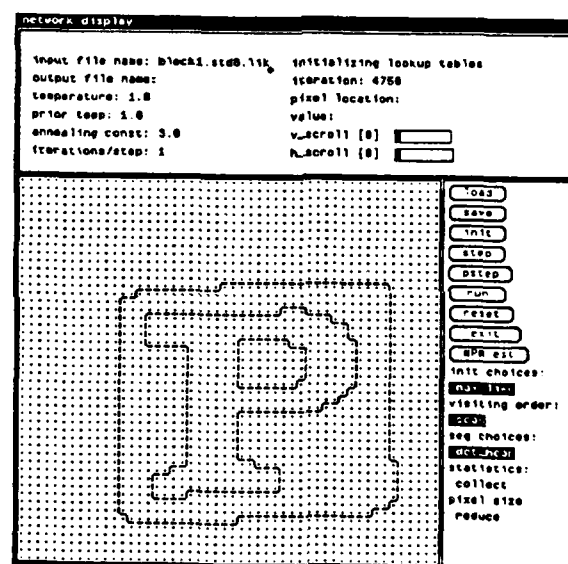
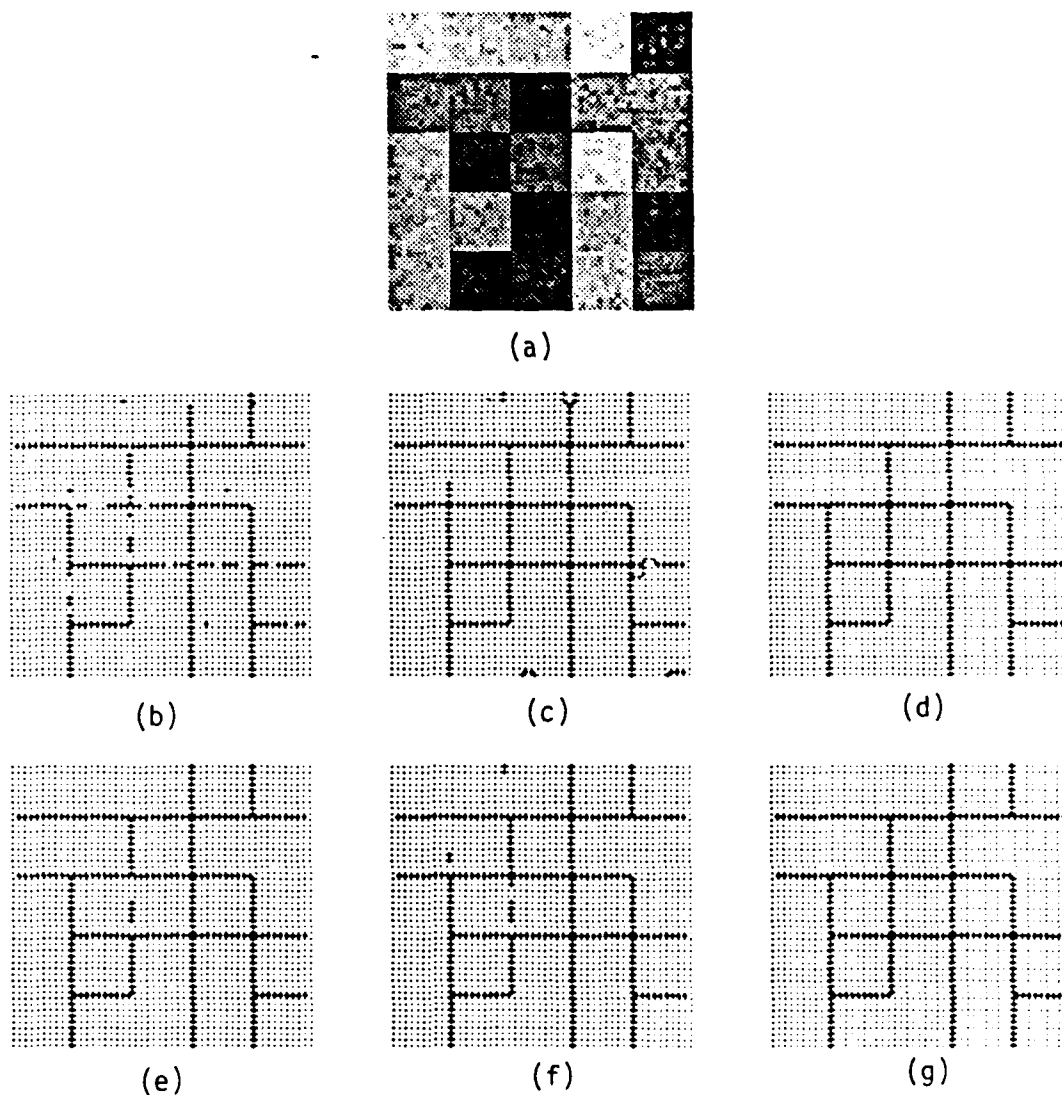
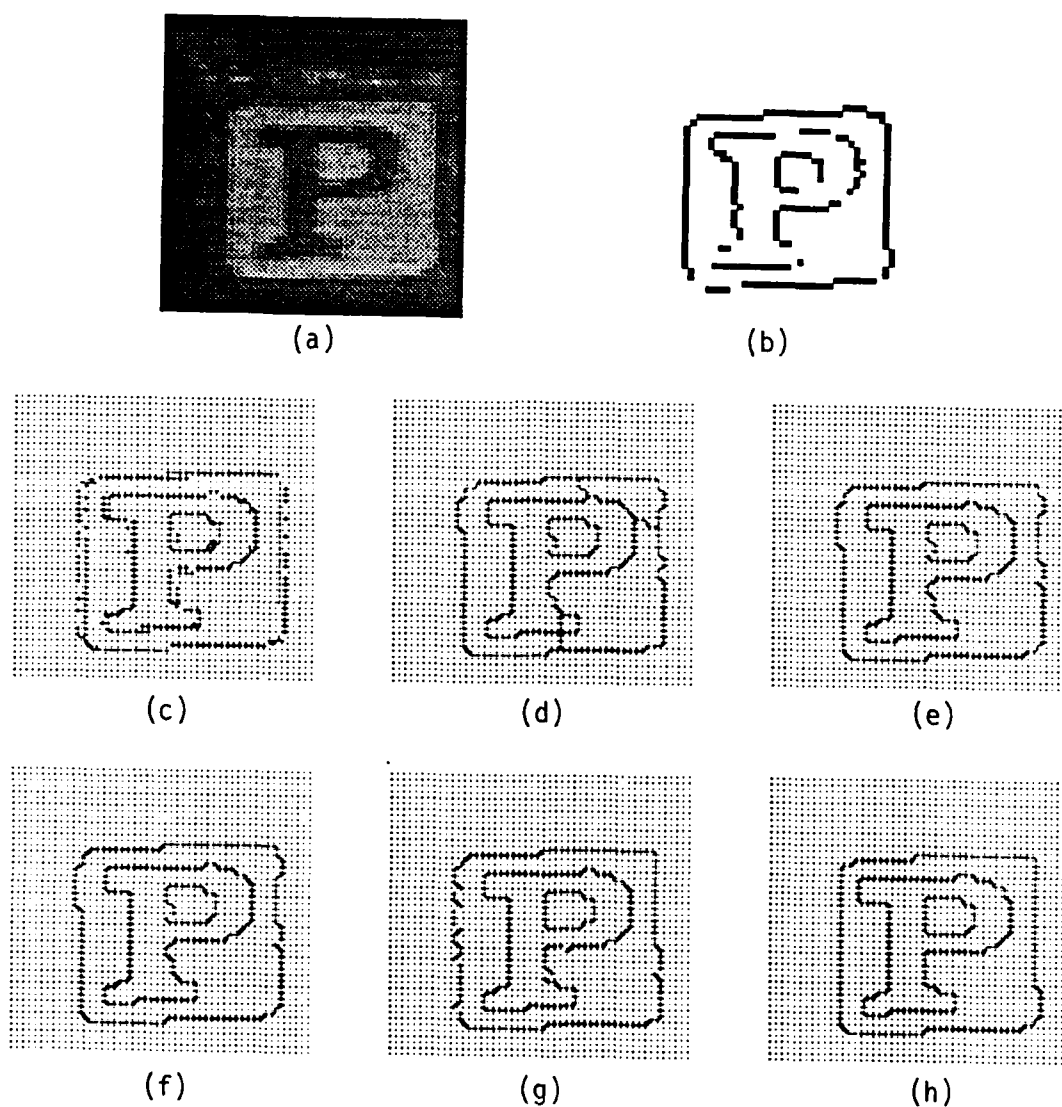


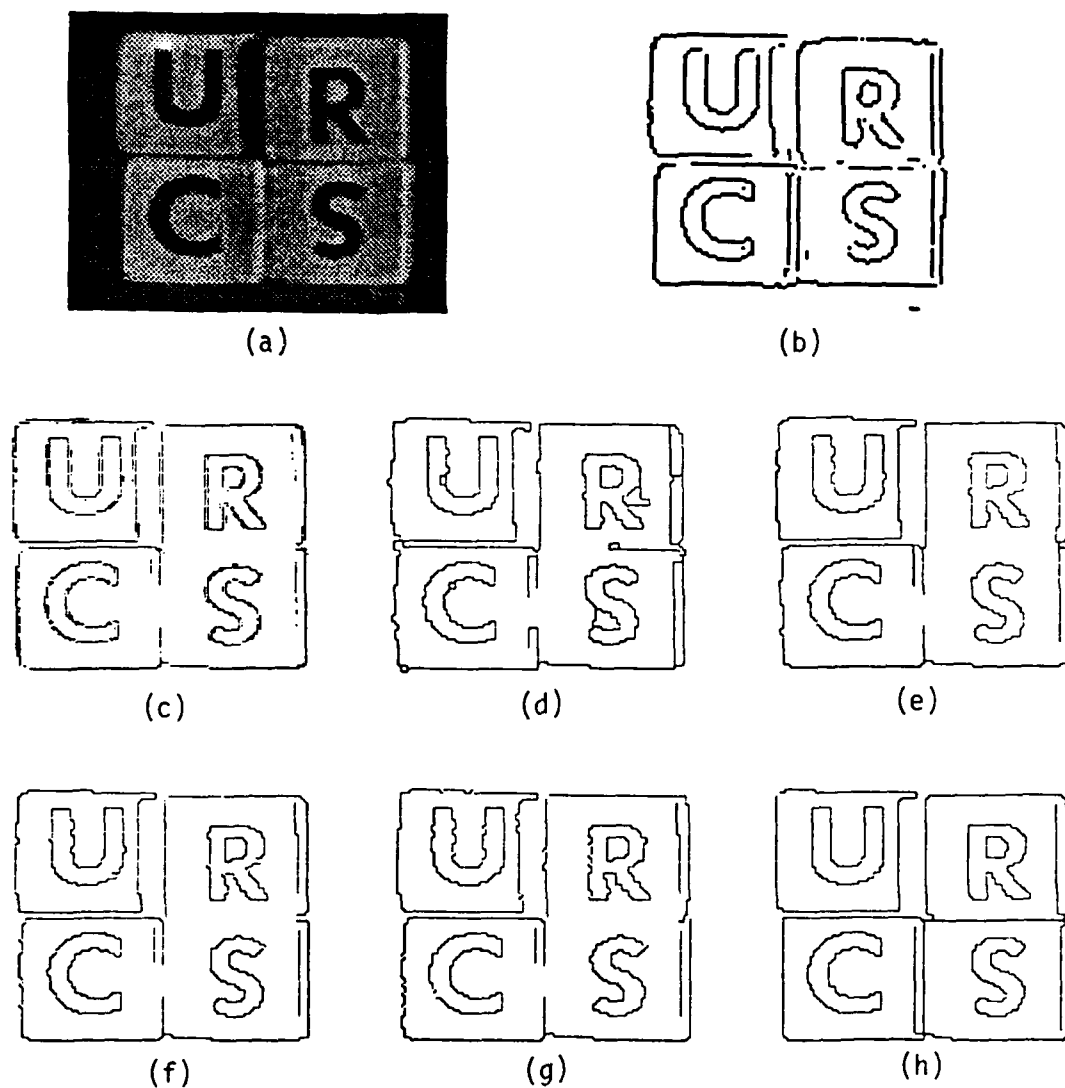
Figure 6: An interactive general purpose MRF simulator.



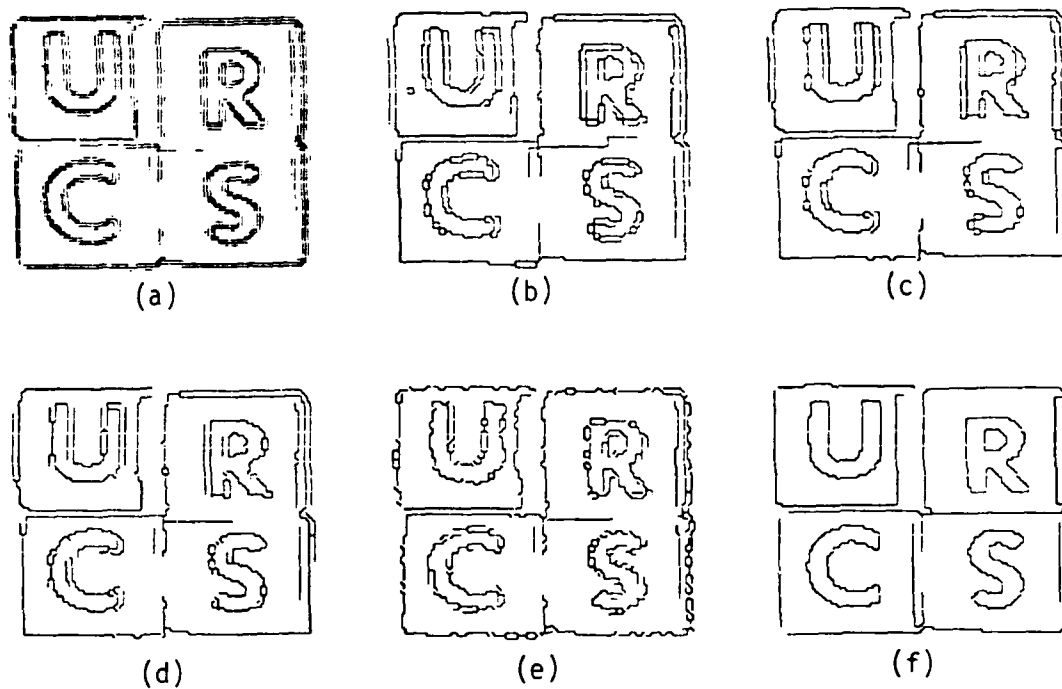
**Figure 7: Experiment set. (a) Synthetic  $50 \times 50$  "checkerboard" image corrupted by independent Gaussian noise, mean 0, standard deviation 16.0. (b) TLR configuration (c) Stochastic MAP estimate. (d) Stochastic MPM estimate. (e) DIR (scan-line visiting order) MAP estimate. (f) DIR (random visiting order) MAP estimate. (g) HCF result.**



**Figure 8: Experiment set.** (a) Natural  $50 \times 50$  image of wooden block. (b) Thinned and thresholded output of  $3 \times 3$  Kirsch operators. (c) TLR configuration. (d) Stochastic MAP estimate. (e) Stochastic MPM estimate. (f) DIR (scan-line visiting order) MAP estimate. (g) DIR (random visiting order) MAP estimate. (h) HCF result.

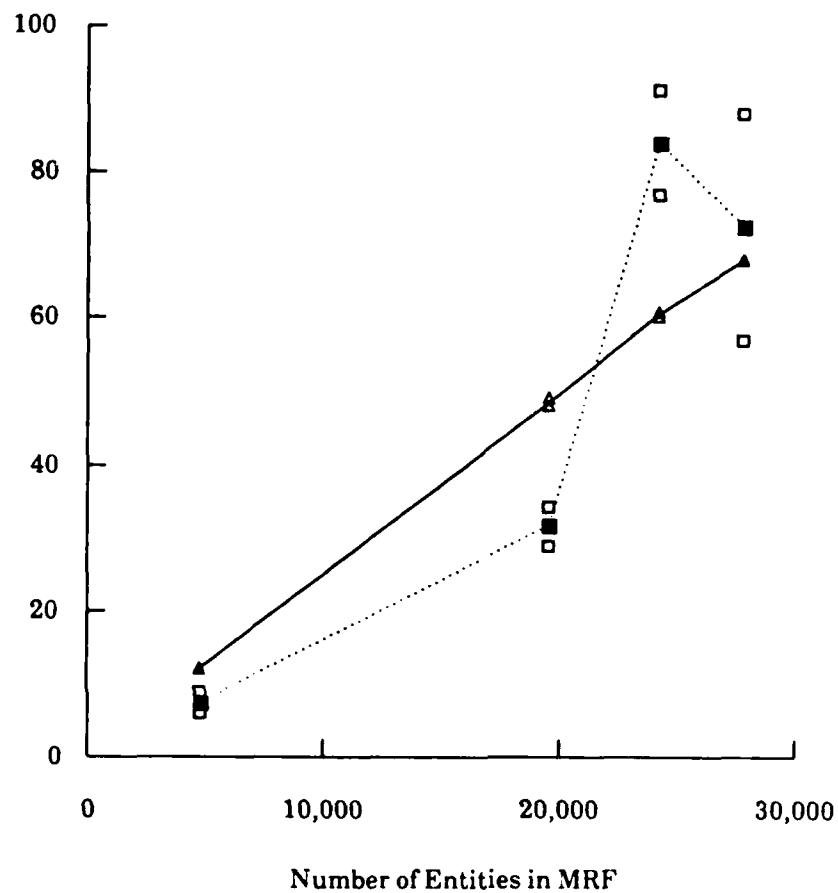


**Figure 9:** Experiment set. (a) Natural 100x124 image of four plastic blocks. (b) Thinned and thresholded output of 3x3 Kirsch operators. (c) TLR configuration. (d) Stochastic MAP estimate. (e) Stochastic MPM estimate. (f) DIR (scan-line visiting order) MAP estimate. (g) DIR (random visiting order) MAP estimate. (h) HCF result.



**Figure 10: Experiments with incomplete edge model - original image in Fig. 9a. (a) TLR configuration. (b) Stochastic MAP estimate. (c) Stochastic MPM estimate. (d) DIR (scan-line visiting order) MAP estimate. (e) DIR (random visiting order) MAP estimate. (f) HCF result.**

Run Time (sec)



HCF:     $\triangle$  Individual     $\bullet$  Average  
 DIR:     $\square$  Individual     $\blacksquare$  Average

**Table 1: Timing Test Results.** The HCF and DIR algorithms are each run on two images of the same size, for four image sizes. Individual and average run-times are shown.

END

DATE

FILMED

MARCH

1988

DTIC